The University of Hong Kong

Final Report (Individual)

Final Year Project: Find A Seat App

Written by: Lai Cheuk Wing (UID: 3035189465)
Groupmate: Wong Sun Day (UID: 3035277626)
Supervisor: Dr. Anthony Tam
Date of submission: 14 Apr, 2019
Abstract

This report presents the methodologies and results of the final year project “Find A Seat App” by the team formed by Lai Cheuk Wing and Wong Sun Day. The project aims at providing an application that can detect in real-time the occupancies of the seats in HKU Main Library. For the purpose of this individual project report, only the computer vision approach is discussed.

Different methodologies were required in 4/F and 3/F of the library. For 4/F that contain fixed desks with dividers, images were obtained from mounting cameras right above a block of tables. Using the pre-defined table space and seat space, Canny edge detection was performed to look for the number of edge pixels within the table. Human detection was also performed to look for persons within the seat space. For 3/F that contain freely-moved tables, images were obtained from multiple existing cameras from a top-down perspective. After performing rectification on the image, the tables were detected from a customized table detection model. To know whether the table is being used by a person or not, a table-level coordinate system was set up to find the relative distance between elbows and table edges. PoseFlow by AlphaPose, a pose estimation model, was used to extract the location of elbows. The system also looked for hogging based on the area of white contours. Lastly, results from multiple cameras were merged and sent to the webserver. The accuracy of the 4/F detection system is high with occasional misses of the Seated status. On the other hand, the accuracy of the 3/F detection was hindered by the result of different components.
Acknowledgements

We would like to express our special thanks to our supervisor, Dr. Anthony Tam, who offered the team the opportunity to do the project “Find A Seat App”. We would also like to thank Dr. Kenneth Wong who gave us advice and help regarding our difficulty in computer vision, and the HKU Library, especially their representative, Jenny, who gave us so much support throughout the project. Last but not least, we thank our friends and families who participated in the simulation video taking section us and who gave us their big hands when we are in need.
# Table of Contents

ABSTRACT ............................................................................................................................. I

ACKNOWLEDGEMENTS ....................................................................................................... II

LIST OF FIGURES .................................................................................................................. V

LIST OF TABLES .................................................................................................................... IX

ABBREVIATIONS .................................................................................................................. X

CHAPTER 1 INTRODUCTION ................................................................................................... 1

1.1 BACKGROUND ................................................................................................................ 1

1.2 MOTIVATION ..................................................................................................................... 1

1.3 RELATED WORK .............................................................................................................. 1

1.4 SCOPE AND DELIVERABLES ......................................................................................... 4

1.5 OUTLINE .......................................................................................................................... 5

CHAPTER 2 METHODOLOGY .................................................................................................. 6

2.1 OVERVIEW ON 4/F AND 3/F ......................................................................................... 6

2.1.1 4/F ................................................................................................................................. 6

2.1.2 3/F ................................................................................................................................. 6

2.1.3 Architecture ................................................................................................................ 7

2.2 METHODOLOGY ON 4/F .............................................................................................. 7

2.2.1 Video source and camera model ................................................................................ 7

2.2.2 Predefine tables and seat areas .................................................................................. 9

2.2.3 Human Detection with pre-trained detection model .................................................. 10

2.2.4 Edge Detection to look for objects on table ............................................................... 11

2.2.5 Status Deduction ....................................................................................................... 12

2.3 METHODOLOGY ON 3/F .............................................................................................. 14

2.3.1 Video source and processing ..................................................................................... 14

2.3.2 Human Detection with AlphaPose and seat status deduction by coordinate distance of tables and persons ................................................................. 16

2.3.3 Standing Filter ............................................................................................................ 18

2.3.4 Table Detection by custom training .......................................................................... 19

2.3.5 Masking and hull shape comparison to look for hogged tables .................................. 19

2.3.6 Status Deduction ....................................................................................................... 21

2.3.7 Consolidate results from multiple cameras ............................................................... 22

2.4 ABANDONED IDEAS .................................................................................................... 23

2.4.1 Stereo Camera ........................................................................................................... 23

CHAPTER 3 RESULT AND FINDINGS .................................................................................... 25

3.1 4/F ..................................................................................................................................... 25

3.1.1 Human Detection with pre-trained detection model .................................................. 25

3.1.2 Canny edge detection to look for hogged tables ......................................................... 26
3.3.3 Overall Performance........................................................................................................27
3.2 3/F ......................................................................................................................................34
  3.2.1 Table detection by custom training ..............................................................................34
  3.2.2 Masking and hull shape comparison to look for hogged tables ................................39
  3.2.3 Standing filter with Random Forest Classifier ..........................................................42
  3.3.4 Consolidate results from multiple cameras .................................................................44
  3.3.5 Overall performance ....................................................................................................46

CHAPTER 4: EVALUATION / DISCUSSION .............................................................................56
  4.4.1 Reliability ....................................................................................................................56
  4.4.2 Scalability ...................................................................................................................56
  4.4.3 Adaptability ................................................................................................................57
  4.4.4 Cost ..............................................................................................................................57

CHAPTER 5: FUTURE IMPROVEMENT AND RECOMMENDATIONS ....................................59

CHAPTER 6: CONCLUSION ....................................................................................................60

REFERENCES ..........................................................................................................................61
List of Figures

Figure 1.3.1  SMU Library heatmap ................................................................. 2
Figure 1.3.2. Standard deviations of object occupancy and people occupancy ........ 2
Figure 1.3.3. OccupEye product photo .......................................................... 3
Figure 1.3.4. Live availability of desks on The University Of Edinburgh library’s website ..... 3
Figure 1.3.5. GridEye installation and coverage ............................................. 4
Figure 1.3.6. Study Hunt App User Interface .................................................. 4
Fig 1.4.1: fixed environment on 4/f (Left) and movable environment on 3/f (Right) ........ 4
Fig 2.1.3.1 Designed Architecture of the vision system on the 3/F and the 4/F ............ 7
Fig 2.2.1.1 Study desks on the aisle of 4/F ..................................................... 8
Fig 2.2.1.2 Video capture of the camera mounted right above a block of desks ............ 8
Fig 2.2.1.3 The additional compact camera was mounted onto the ceiling ................ 9
Fig 2.2.1.4 Closer look of the camera ............................................................ 9
Fig 2.2.2.1 Pre-defined seating space (yellow) and desk space (white) of a block of tables ... 10
Fig 2.2.3.1 Result of human detection on a block of tables with pre-defined seating space and table pixels (white dots) ................................................................. 10
Fig 2.2.3.2 The human detection model detecting the person on the cover of the book ......11
Fig 2.2.4.1 Original desk image (left) and edge pixels (right) of an empty table .......... 11
Fig 2.2.4.2 Original desk image (left) and edge pixels (right) of 3 hogged tables .......... 12
Fig 2.2.5.1 The flowchart of how a status was derived on 4/F ............................ 13
Figure 2.3.1.1. Location of surveillance cameras on the 3/F compartment. The camera at the bottom is not working ................................................................. 14
Figure 2.3.1.2. Video capture of surveillance cameras on the 3/F compartment ............ 14
Fig 2.3.1.3 Camera on the right that ignores pixel values with y < 200 to mimic the room dividers ................................................................. 15
Fig 2.3.1.4. 2 snapshots of the calibration process using the calibration chessboard ..... 15
Fig 2.3.1.5. Video capture before rectification (left) and after rectification (right) ....... 15
Fig 2.3.1.6 Video capture before rectification (left) and after rectification (right) ....... 15
Fig 2.3.2.1. Calibration chessboard is placed on a table to set up the coordinate system at the table level. The corners of each square are detected. ................................. 17
Fig. 2.3.2.2. Coordinate system projected from the calibration chessboard on the table. (0,0) starts at the bottom left corner of the calibration chessboard ..................................... 17
Fig 2.3.2.3 Table borders and elbows (white points) obtained their coordinate system. .....18
Figure 2.3.5.1. The above images shows the steps of the masking on an empty table .......20
Figure 2.3.5.2. The above images shows the steps of the masking on an occupied table that has 1 connected white contour. ......................................................... 20
Fig 2.3.5.3. The above images shows the steps of the masking on an occupied table that has multiple white contours while the largest contour is over 75% of the width of the bounding box .................................................................................................................... 20
Fig 2.3.5.4. The above images shows the steps of the masking on an occupied table that has multiple white contours while the largest contour is below 75% of the width of the bounding box .................................................................................................................... 20
Fig 2.3.6.1. The flowchart of how a status was derived on 3/F ............................ 22
Fig 2.4.1.1 Parallel placement of 2 identical cameras to work as a stereo camera .................................................................23
Fig 2.4.1.2. Steps of stereo matching ........................................................................................................................................24
Fig 2.4.1.3. Disparity map of 3/F ...............................................................................................................................................24
Fig 2.4.1.4. 3D image of 3/F produced from the disparity map ....................................................................................................24
Fig 3.1.1.1. Correct human detection outputs from the model ..................................................................................................25
Fig 3.1.1.2 Incorrect human detection outputs from the model ................................................................................................26
Fig 3.1.2.1. Table status result and number of edge pixels within the table (left) and the original image (right) ..............................................................................................................................................27
Fig 3.1.2.2. top right desk: The number of edge pixels is 133 when a pen is put on the table...27
Fig 3.3.1.1 Youtube videos of 4/F detection output ...................................................................................................................29
Fig 3.3.1.2 The top right desk has a status of Hogged, but is wrongly identified as Seated (4f_vid1, 01:12) .........................................................................................................................................................30
Fig 3.3.3.4 Correct Seated status at Top left table: hands off the table (v2_evaluation 00:06)...31
Fig 3.3.3.5 Correct Seated status at Top left table: lying on the table (4f_vid1 01:05)..................................................................31
Fig 3.3.3.6 Correct Seated status at Bottom right table: standing up (v2_evaluation 00:34)..............................................................31
Fig 3.3.3.7 Correct Seated status at Bottom right table: turning sideways (v2_evaluation 01:37) ..........................................................31
Fig 3.3.3.8 Correct Seated status at Top left table: using laptop (v2_evaluation 01:11).................................................................32
Fig 3.3.3.9 Correct Seated status at bottom right table: playing on the phone (hog_evaluation 02:45) ........................................................................................................................................................................32
Fig 3.3.3.10 Correct hogged status at top right table: Hogged with laptop, top left table: hogged with jacket (4f_vid1 01:16) ......................................................................................................................................................32
Fig 3.3.3.11 Correct hogged status at top right table: hogged with paper (hog_evaluation 03:52) .............................................................................................................................................................................33
Fig 3.3.3.12 Correct hogged status at Top right table: hogged with a small white object (pen) (hog_evaluation 03:00) .........................................................................................................................................................33
Fig 3.3.3.13 Correct hogged status at Bottom right table: hogged with a small black object (hog_evaluation 00:59) ..............................................................................................................................................................33
Fig 3.3.3.14. Correct hogged status at Top right table: hogged with a white book (hog_evaluation 03:31) ........................................................................................................................................................................33
Fig 3.3.3.15 Correct empty statuses for empty tables (4f_vid1 00:23) .........................................................................................34
Fig 3.3.3.16 Correct empty status at bottom right table: people passing by (4f_vid1 01:43).............................................................34
Figure 3.2.1.1 Overall mAP at different training iterations in the first custom training. .................................................................34
Figure 3.2.1.2 Overall mAP at different training iterations in the final custom training. ..........................................................35
Figure 3.2.1.3. mAP of objects in large size (left), medium size (middle) and small size (left) at different training iterations in the final custom training..................................................................................36
Fig 3.2.1.4 Detection result (left) and ground truth (right) ........................................................................................................36
Figure 3.2.1.5. are the output examples of the model on an unseen video captures of 3/F. ............................................................37
Fig 3.2.1.6. Video capture of camera 1 (left) and the detection result (right) ..................................................................................37
Fig 3.2.1.7. Video capture of camera 2 (left) and the detection result (right) ..................................................................................37
Fig 3.2.1.8. Video capture of camera 3 (left) and the detection result (right) ..................................................................................37
Fig 3.2.1.9 Video capture (left) and inaccurate detection result (right) of a table being blocked by a person in the middle..............................38
Fig 3.2.1.10. Video capture (left) and accurate detection result (right) of a table when the person has walked past the table ................................................................. 38
Fig 3.2.1.11 Video capture (left) and missing detection result (right) of a table being blocked by multiple persons ................................................................. 38
Fig 3.2.1.12 Video capture (left) and accurate detection result (right) of a table being blocked by multiple persons ................................................................. 39
Fig 3.2.1.13 Multiple detection boxes over cropped tables placing close together .................... 39
Fig 3.2.2.1 Decision tree for unblocked hogged tables produced from the small amount of training data. Value = [Hog, Seated] ................................................................. 40
Fig 3.2.2.2 Decision tree for unblocked hogged tables after manual adjustment ........................ 41
Fig 3.2.2.4 Decision tree for blocked hogged tables ................................................................ 42
Fig 3.2.2.5 Original image of the blocked table (left) and the white contours (right) .................... 42
Fig 3.2.3.1 Accuracy and F1 score of the sit-stand random forest classifier when using different n_estimators and maximum depth .................................................. 43
Fig 3.2.3.2 Examples of correct identification from the classifier, True = sitting, False = standing ........................................................................................................ 44
Fig 3.2.3.3 Examples of incorrect identification from the classifier, True = sitting, False = standing ........................................................................................................ 44
Fig 3.3.4.1. Original images from the camera 1 (left), camera 2 (middle) and camera 3 (right) 45
Fig 3.3.4.2 Visualization of Status result from system ........................................................ 45
Fig 3.3.4.3. Consolidation of output across cameras with tables across cameras overlapping with each other ................................................................. 45
Fig 3.3.4.4 Consolidation of tables and statuses ................................................................ 46
Fig 3.3.4.5 Youtube videos of 3/F detection system ............................................................. 46
Fig 3.3.5.2 Multiple persons sitting around a table (blue) and sitting with different angles (purple) (1mar_084500_evaluation 04:19) ................................................................. 47
Fig 3.3.5.3 Person with lifting arms (pink) (9apr_0083001_v2_evaluation 02:02) ................... 48
Fig 3.3.5.4 Person sitting with different angles (pink) (9apr_0083001_v2_evaluation 03:43) .... 49
Fig 3.3.5.5 Person sitting and facing backward (pink) (9apr_0083001_v2_evaluation 01:10) .... 49
Fig 3.3.5.6 Table is hogged with complicated pattern (1mar_084500_evaluation 04:19) .......... 50
Fig 3.3.5.7. Hogged seats with a laptop (blue) and multiple items (pink) (9apr_0083001_v2_evaluation 02:02) ........................................................................ 50
Fig 3.3.5.8 Hogged seats being blocked by a person (pink) (9apr_0083001_v2_evaluation 02:27) ........................................................................................................ 50
Fig 3.3.5.9. Empty seat with a chair blocking the view (9apr_0083001_v2_evaluation 01:53) 51
Fig 3.3.5.10 Empty seat with a chair blocking the view (9apr_0083001_v2_evaluation 03:23) 51
Fig 3.3.5.11 Empty seat with a chair blocking the view (9apr_0083001_v2_evaluation 01:53) .51
Fig 3.3.5.12 Seated table partly outside the camera’s view is labelled as Empty .................... 52
Fig 3.3.5.13 Occupied tables were being labelled as Hogged (1mar_084500_evaluation 01:55) 53
Fig 3.3.5.14. Alphapose was not able to detect two persons in the image .............................. 54
Fig 3.3.5.15. The empty table with a white color board behind it was wrongly labelled as hogged. (9apr_0083001_v2_evaluation 01:53) ................................................................. 55
Fig 3.3.5.16 The original image of table with a white board at the back (left) and contours (right) with estimated table shaped outlined in yellow.
List of Tables

Table 3.1.2.1. Number of edge pixels for respective objects observed during the testing ........27
Table 3.3.3.1 Number of correct classification in 3 testing videos ..................................29
Table 3.3.3.2 Number of incorrect classification in 3 testing videos .................................29
Table 3.3.3.3. Use cases for 4/F detection system .................................................................30
Table 3.3.5.1. Use cases of testing 3/F detection system .......................................................47
Table 4.4.5.1 Time required in each components of 3/F detection system .........................58
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>Computer Vision</td>
</tr>
<tr>
<td>HKU</td>
<td>The University of Hong Kong</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>mAP</td>
<td>Mean Average Precision</td>
</tr>
<tr>
<td>PIR</td>
<td>Passive Infrared</td>
</tr>
</tbody>
</table>
Chapter 1 Introduction

1.1 Background

HKU libraries provide study places for students and staffs. Serving more than 125,000 registered library users with 2957 study places [1], the libraries are struggling to keep up with the huge demand for seats. Many students endured a difficult and time-consuming search for an empty chair, particularly during the examination period. While some seats allow prior booking via the HKU Portal, some are first-come-first-serve and are under keen competition. Furthermore, hogging seats with personal belongings for a long time are some notorious acts that are very common in the libraries. Since it is hard to look for a seat, visitors have to browse in every room from floor to floor. An app that tells the occupancy of seats availability and location of empty seats would replace the time-consuming search by a few taps on the phone.

1.2 Motivation

The HKU Main Library has been receiving complaints that it is tough to find a seat during the examination period. In hope of improving the users’ experience and to reduce the nuisance caused by wandering visitors, this project pursues the digitalization of seat occupancy. A web mobile application that indicates which seat is ‘Empty’, ‘Hogged’, or ‘Seated’ effectively tells the user where they should head to directly.

This project explores the use of image recognition and sensors in providing near real-time seat occupancy information and delivers the service on an application that is accessible from a visitor’s smartphone.

1.3 Related Work

Singapore Management University (SMU)

In SMU’s library, they provide live occupancy density of different areas. It is represented through a heatmap, where the more red the more dense the area while the greener the less crowded (See Figure 1.1). Wifi devices are placed in different corners of the room to monitor the network data flow to predict the density of library users in that area.
In another research led by Nguyen et al. from majorly SMU in 2013, they have discovered using capacitive sensor to detect the hogging state of the table, where hogging is defined by the author as “impromptu reservation” [2]. It compares the capacitance detected when there are people or objects in its proximity with the background capacitance to deduce occupation. One significant finding by the research team is the difference in the degree of signal fluctuation when there are human nearby and when there are only objects in its surroundings. As depicted in the Figure 1.3.2, the orange line which denotes the signal outputs when the seat is occupied by human fluctuates substantially while the blue line which denotes the detected capacitance when the seat is occupied by objects remains stable. The researchers leveraged the finding to further determine whether the seat is occupied by human or objects. By Jan 2017, the hogging detection system had been deployed across 90 seats in SMU library and achieved 91.2% accuracy [3].

The University Of Edinburgh

OccupEye is a commercial product that is capable of detecting seat vacancy (See Figure 1.3.3). The major component is the passive infrared sensor (PIR) inside that can identify human motion in its proximity. The sensor is installed on each desks on their Main Library Lower Ground Floor. The data collected is then reflected on their website which displays live availabilities of the desk (See Figure 1.3.4).
University College of London (UCL)

A team of 5 undergraduates in UCL accomplished a project called “Study Hunt” in which they used GridEye, a thermal camera, to monitor the real-time vacancy of the seats in their Library [4]. The cameras are mounted on the ceiling (See Figure 1.3.5) where each of them looks over 4 seats. Temperature of every point within its detection area (the blue plane) will be measured. Because the location of the seats are known in advance, if there are human heat in the area supposed to be a seat, it can be further deduced that it is occupied by someone. The user interface of the app is shown in Figure 1.6. At the bottom, the availabilities of four seats around the table can be seen, where red means occupied, green means empty.
1.4 **Scope and Deliverables**

The team will deliver a web application that displays whether a seat is ‘empty’, ‘hogged’, or ‘seated’ on a 2D floor plan in almost real-time with the maximum latency of 1 minute. The app takes care of both fixed and movable environments in the Library. A study area on 4/F with 28 non-movable desks is chosen to be our testing site. On the other hand, for movable settings, one of the rooms on 3/F with approximately 30 movable tables and around 52 chairs is selected (See Figure 1.4.1). In addition, the size of the room is 10.1m x 14.2m.

The app also enables library administrators to set-up the detection system in a new room.

This project covers the seat occupancy information of a few seating areas of the HKU Main Library. The seating areas come in different furniture settings and compartments. It is used to demonstrate the feasibility of implementing the system in a variety of settings.
This project explores two different approaches of showing table statuses: using computer vision and IoT. This individual report focuses on the computer vision approach while another groupmate writes about the IoT approach.

1.5 Outline

The paper is organized into six chapters. Chapter one has offered the background and the purpose of the project. Chapter two presents the methodologies used for 4/F and 3/F. Chapter three presents results and findings in details. It illustrates the use cases used for testing and provides evaluation of them. Chapter four illustrates the reliability, scalability adaptability and cost of the systems. Chapter five highlights the future work. Chapter six highlights the key points of the previous chapters and concludes the paper.
Chapter 2 Methodology

2.1 Overview on 4/F and 3/F

2.1.1 4/F
On 4/F, cameras were mounted above every block of tables. The pixel coordinates of table corners and seat corners on the image were pre-defined because the location of the tables are fixed already. The hogging status of a seat was determined by finding the amount of edges on the table surfaces. The seated status was determined by detecting the existence of persons in the pre-defined seat area using a pre-trained object detection weights.

Below are the implementation steps:
1. Obtain images of the room
2. Perform object detection to look for persons within the predefined pixel coordinates of seat corners
3. Perform edge detection to look for objects on the table
4. Deduce the status

2.1.2 3/F
Unlike 4/F, wheeled tables on the 3/F are intended to be moved freely within the compartment. Therefore, the location of tables, on top of persons, were found using object detection. The hogging status was determined by a mask and comparison with its hull shape. To find out if a seat is in seated status, the relative distance between persons and tables was discovered by setting up a coordinate system at the table level, placing the pixel values of tables and elbows of the person at the same plane. The elbows were located with PoseFlow from AlphaPose, a pose estimation model. The seated status is deduced if the coordinate distance between the table and the person is within a threshold.

At last, the floorplan was constructed from results coming from different cameras.

Below are the implementation steps on 3/F:
1. Obtain images of the room
2. Perform table detection with customized training
3. Set up the table-level coordinate system
4. Human and elbow detection with AlphaPose and seat status deduction by coordinate distance
5. Filter out persons who are standing
6. Masking and hull shape comparison to look for hogged tables
7. Deduce the status
8. Consolidate results from multiple cameras and merge overlapped tables
2.1.3 Architecture

The designed architecture of the system on the 3/F and the 4/F is illustrated in the figure X.X below.

Fig 2.1.3.1 Designed Architecture of the vision system on the 3/F and the 4/F

Fig 2.1.3.1 shows the designed architecture of the vision system on 3/F and the 4/F. Cameras would send frames wirelessly to the GPU machine. The GPU machine processes the images and post the result to the server. The server updates the database upon receiving the data. The server requests for the most updated record from the database and display the result onto the webpage.

The setup used during testing is simplified. On the 4/F, only one camera is used for demonstration purpose. Frames were wirelessly transmitted to the GPU machine in real-time. Afterwards, it follows the data flow illustrated in Fig 2.1.3.1. On the 3/F, video frames were not obtained in real-time but were processed offline as video files.

2.2 Methodology on 4/F

2.2.1 Video source and camera model

The isle on 4/F is placed with 9 blocks of study desk that are separated by dividers as shown in Fig 2.1.1.1. These dividers blocked the table surface from existing cameras and it was impossible to know whether some seats are seated or hogged.
To work around the block of view constraint brought by the table dividers, the team decided to mount a compact camera right above a block of tables shown in Fig 2.2.1.2. Each camera is only responsible for one block of desks. In other words, 9 cameras are needed to cover the entire room. This project only works with one camera for demonstration purpose. An image from the camera is shown in Fig 2.2.1.3.
The camera is bought from a company called Yuen Cheng Advanced Technology Electronics Company on Taobao.com. The model name of the camera is A9. The camera has a resolution of 1080p with a 150 degrees view. It is powered by a rechargeable battery that can sustain for 60 minutes without connecting to a power socket. It has a WiFi hotspot built in and live video was streamed wirelessly. For the convenience of doing the project, the team made use of the SD card to obtain videos. Live video was usually used when the team tests the implementation in the actual environment.

2.2.2 Predefine tables and seat areas

Each camera is responsible for a fixed block of tables that consist of 4 seating spaces. The block of tables was not designed for visitors to move around. Therefore, an assumption is held that the location of the block under its assigned camera is fixed.

The corner pixel values of the entire seating space and the desk space are predefined into the system manually. Fig 2.2.2.1 shows the pre-defined table and seat areas. This step defines the boundary of a table that would be processed for edge detection to see if it is occupied by objects in the later stages. The results from human detection would also be checked to see if they overlap with the seating space. This means how they were configured affect the output of the system significantly.
2.2.3 Human Detection with pre-trained detection model

The goal to find if there are any persons overlapping with the pre-defined seating space. As Human detection is one of the most common detection tasks, pre-trained weightings are widely available. Faster RCNN Inception v2 is the network chosen for human detection. A set of pre-trained weightings trained on COCO image set is used [5].

A Faster RCNN Inception v2 model was used for its high accuracy. When choosing among other networks, the speed and accuracy are considered. Faster RCNN in general takes longer time than SSD networks due to its larger size but is compensated with a higher accuracy.

The claimed mAP of the pre-trained weights on the COCO dataset is 28. The claimed speed of Faster RCNN Inception v2 is 58 ms. The GPU machine the team used for this project is NVIDIA GeForce 930M with 4GB of memory. Assigning 30% of GPU memory for the process, the actual running time for processing an image on the machine is around 0.41 seconds.
The size of the person detected was checked because any pictures with a person placed on the table can also be detected. The detected person is filtered away if the 4 corners of the person bounding box lie within the table, the person’s width is less than 0.7 of table width, or height less than 0.7 of table height. Fig 2.2.3.2 shows an unwanted human detection which a book has an image of a man on the cover.

![Image](image1.png)

**Fig 2.2.3.2** The human detection model detecting the person on the cover of the book

### 2.2.4 Edge Detection to look for objects on table

A box filter was applied to the table area to remove noise. Then, Canny edge detection was performed to look for edges of objects on the table. The number of detected edge pixels was used as the decision metric. Fig 2.2.4.1 shows a list of edge pixels for empty and hogged tables.

![Image](image2.png)

**Fig 2.2.4.1** Original desk image (left) and edge pixels (right) of an empty table
Counting the number of edge pixels in the predefined table area, a threshold was applied to filter those with more edges and label them as hogged.

### 2.2.5 Status Deduction

Fig 2.2.5.1 shows the flow of the decision-making process for 4/F.
As shown in Fig 2.2.5.1, the decision making of seats on the 4/F went through the following steps. Firstly, the system checks if the seating space overlaps with any person. If yes, the system checks to see if the person is within a reasonable height and width. If the person is not overly small, the seat is labelled as Seated.

If the seating space is not overlapping with any person, or such person is unreasonably small, the system proceeds to check if the table is hogged. It uses the result from canny edge detection and checks the number of pixels within the predefined table. If number of edge pixels exceeds the value 100, the seat is considered Hogged. Otherwise, the seat is considered Seated.
2.3 Methodology on 3/F

2.3.1 Video source and processing

The compartment on 3/F is equipped with 4 cameras. Existing cameras were used as image source because they cover the room well and no additional installation of cameras are required. However, one of the cameras was not working properly throughout the entire process of this project and no video could be obtained. As a result, all work in the following sections only consists of frames from 3 of the cameras.

Note that the camera on the right is only responsible for capturing until the room dividers. For the video clips the team obtained, there were times when the divider was removed in the room. In this case, only scene objects that have a y pixel value larger than 200 is considered.

Figure 2.3.1.1. Location of surveillance cameras on the 3/F compartment. The camera at the bottom is not working.

Figure 2.3.1.2. Video capture of surveillance cameras on the 3/F compartment
The camera views in Fig 2.3.1.2 and 2.3.1.3 suffer from barrel distortion. Rectification is required to establish an accurate table-level coordinate system which details are discussed in Section 2.3.2. The matrix to be used rectification was calculated by chessboard calibration. Since the cameras were mounted at the ceiling and there is difficulty in calibration the lens within a short distance, a large calibration chessboard of 80cm x 100cm was used. With the camera matrix in hand, images from the cameras were rectified before passing them to the next step. The rectifications are shown in Fig 2.3.1.5 – 2.3.1.6.
2.3.2 Human Detection with AlphaPose and seat status deduction by coordinate distance of tables and persons

Whether the detected table is in Seated status depends on its distance with detected persons. The challenge is that under the perspective of the camera, the depth of objects are lost. It becomes ambiguous when a person overlaps with multiple tables at the same y-axis on the image.

In this project, a coordinate system was set up at the table plane to find the relative distance between persons and tables. Making use of the observation that the elbows of a seated person are usually placed at a similar height as the table, the system collects all elbow pixels and table corner pixels in the image that were assumed to be lying on the same plane. The team use Poseflow from Alphapose, a pose detection system powered by PyTorch that returns the location of body joints under a non-commercial license [6]. The elbow locations were used as the reference point of the person at the table height. Fig 2.3.2.1 shows an example output of Alphapose. The location of body parts and joints are returned.

![Fig 2.3.2.1 Example output of Alphapose](image_url)

To set up the coordinate system, a 8x7 calibration chessboard is placed on a table as shown in Fig 2.3.2.2. The pixel values of the chessboard corners were then mapped to coordinate values from (0,0) to (8,7). The mapping was then extended to the entire image to set up a coordinate system in the entire room. When finding the relative distance between a person and a table, the system first maps the pixel values of related elbow and table pixels onto the coordinate system. The Euclidean distance between the coordinate values of the elbow and table were then obtained and used for comparison.
Fig 2.3.2.2. Calibration chessboard is placed on a table to set up the coordinate system at the table level. The corners of each square are detected.

Fig. 2.3.2.3. Coordinate system projected from the calibration chessboard on the table. (0,0) starts at the bottom left corner of the calibration chessboard.

A square on the calibration chessboard is 3.7 cm long and wide, meaning 1 unit of length in the coordinate system represents 3.7 cm in the real world. The goal was to find the person who is within a reasonable distance from the table. A threshold was set that if the person’s elbow is within 22 cm from the table, or 6 units in the coordinate system, the person is regarded as using the table. This means if a person has a coordinate distance less than 6 with a table and that table is closest to that person, the table is
labelled as Seated. There is an assumption made is that a person that only occupy 1 table. One thing to note is that when the table’s coordinates enclose the elbow’s coordinate, the table is also labelled as seated.

Here is an example in Fig 2.3.2.4. Using the coordinate system, the system find the coordinates of elbows and tables shown in the image. For example, the boy on the left has a coordinate value of (34.1, -40.6) on his detected left elbow. The table he is using, which is the one in front of him, has a point that has the coordinate value (29.2, -41.7). The table behind him has a point with coordinate value (53.5, -45.3). The distance between the boy and the front table is at least 5.02 units. The distance between the boy and the back table is at least 19.3 units.

![Fig 2.3.2.4 Table borders and elbows (white points) obtained their coordinate system.](image)

**2.3.3 Standing Filter**

Not all persons detected in the scene are taking up a desk space. The system has to avoid matching a table with a standing person who is walking pass but not using the table. People who are walking pass, in other words not in sitting position, were filtered out by a Random Forest Classifier. Using the body joints locations returned from AlphaPose, a few metrics were chosen for deciding whether a person is standing or sitting:

1. Angle between left shoulder - left hip - left knee
2. Angle between right shoulder - right hip - right knee
3. Angle between left hip - left knee - left ankle
4. Angle between right hip - right knee - right ankle
5. Width-to-height ratio

The above criteria were picked from the intuition that people who are seated bend their hips and knees. A person who is seated is also likely to have a larger width-to-height ratio.
2.3.4 Table Detection by custom training

Table detection is performed on 3/F because their location is not fixed. Faster RCNN Inception v2 network was chosen and trained on an image set of library tables taken on the 3/F. The speed of the model can be referred to section 2.2.3.

Using a pre-trained model on COCO dataset, custom training was first performed on an image set of 100 images for the initial test. Afterwards, another custom training was performed additionally on an image set of 900 images. There was a need for custom training as the pre-trained weightings of the model produces a poor result in detecting the C-shaped tables on the 3/F.

The final training was conducted on a 10:1 split of training and testing data.

2.3.5 Masking and hull shape comparison to look for hogged tables

To determine whether there are any objects on the tables, a few different processes are needed to recover the original shape of table and amount of table surface being occupied.

It is necessary to extract the actual shape of the table from the bounding box for finding how much the surface is blocked by objects. Making use of the advantage that all tables on the floor are white in color, pixels that lie within a predefined range of white color are assumed to be the desk surface. Using this method, books, bags, and other objects of different shapes are filtered out.

Firstly, a median filter was applied to reduce the amount of noise on the table. Next, a bitwise filter was applied to select pixels that lie within a small range of white and light grey. From the masking result, the system extracted the white fragments by looking for contours.

Next, the system produced an estimation of the actual desk shape from the white fragment. The system first check if the largest white contour takes up over 75% of the width of the bounding box. If so, the hull of the largest white contour is used as the actual desk shape. If the largest white contour takes up less than 75%, it is likely other contours also make up the desk surface. The next largest contour will be selected and so on until the width span of the contours reach 75% of the box. The hull of the selected contours will be used as the actual desk shape.

Below are the steps for finding the shape of desk:
1. Apply median filter on the table
2. Bitwise filter to select pixels that are white
3. Find contours from white pixels
4. Check the width of the largest white contour
5. If width of largest white contour >= 75% of the width of the bounding box, its hull is estimated to be the desk shape
6. If width of largest white contour < 75% of the width of the bounding box, the subsequent largest white contour is selected until the width span reaches 75% of the box width. The hull of the selected contours is estimated to be the desk shape
The respective steps are illustrated in Figure 2.3.5.1 – 2.3.5.4. In the figures, the original image of the detected table (left), binary mask result on the table (middle), white contours outlined in blue and desk shape estimation in yellow (right) are shown.

Figure 2.3.5.1. The above images shows the steps of the masking on an empty table.

Figure 2.3.5.2. The above images shows the steps of the masking on an occupied table that has 1 connected white contour.

Fig 2.3.5.3. The above images shows the steps of the masking on an occupied table that has multiple white contours while the largest contour is over 75% of the width of the bounding box.

Fig 2.3.5.4. The above images shows the steps of the masking on an occupied table that has multiple white contours while the largest contour is below 75% of the width of the bounding box.

Afterwards, all information is ready for deducing hogged status. There are two cases the system needs to handle: (1) table is blocked by objects, and (2) table is blocked by a person due to perspective. For case (1), the white contours are fragmented in more irregular shapes. For case (2), if the empty table is blocked by a person due to perspective, the white contours are fragmented but are likely to be smoother on the edges. To avoid misidentifying empty tables of case (2) as hogged, a different process is required.

Firstly, for case (1) where tables are not overlapping with any persons, the system extracts the below features for decision making:

1. Ratio of area of all white contours within estimated table shape to the hull area of the table shape
2. Number of white contours
The intuition is that if the area of white contours within the estimated table shape is less than a certain % of its hull area, there is a high probability that there are a lot of non-white fragments, in other words objects. The second feature is the number of white contours. If a table has a lot of white fragments, it is likely that the table has objects on it.

The system filtered out case (2) tables by checking whether the tables overlap with any persons. Afterwards, the system extracts the below features for decision making:

1. Ratio of the largest white contour to the hull area of the largest white contour
2. Ratio of edges pixels to the perimeter of estimated table shape (outline in yellow in Fig 2.3.5.1 – 2.3.5.4)
3. Number of white contours

It would be meaningless to compare on the entire table shape just like in case (1) because the amount of white contour areas must be smaller. The intuition in this case is that since the table is fragmented by the person blocking the view, the system relied on the shape of the largest contour. If the largest contour area is much smaller than its hull area, there is a high probability that the table is hogged. The system also used the number of edge pixels within the enclosed table shape to deduce amount of objects on the table. The system also looked for the number of white contours. If a table has a lot of white fragments, it is likely that the table is hogged when it is blocked by the person. These features were compared against a threshold and retrieved the table status.

**2.3.6 Status Deduction**

Fig X.X shows the flow of the decision-making process for 3/F.
As shown in Fig 2.3.6.1, the decision making of seats on the 4/F went through the following steps. Firstly, the system checks if the table closest to the person is within 6 units of Euclidean coordinate distance. If yes, this means the table is under suspicion that it is seated. The system then checks whether that person is in sitting position. If yes, the table is labelled as Seated.

If a seated status is not considered, the system proceeds with finding out whether the table is hogged or not. It checks whether the table is overlapping with a person. If yes, the system will make use of features such as the area of the largest white contour and number of white contours. Tables that fulfil the logic will be labelled as Hogged and Empty if otherwise. If the table is not overlapping with a person, the system will make use of other features such as the estimated table shape. Tables that fulfil the logic will be labelled as Hogged and Empty if otherwise.

### 2.3.7 Consolidate results from multiple cameras

To show the floorplan on the web app, the tables from different cameras were consolidated based on the orientation and rotation of the cameras. The table result was flattened out to remove the perspective. Since the cameras have overlapping field of view, the overlapping areas were predefined. Tables appear in
overlapping zones will be consolidated. The hull of the union of the overlapped tables were chosen and the original two tables would be removed.

If the overlapped tables do not have a matching status, the status of the combined table will be assigned under the precedence order of Seated, Hogged and Empty of the origin tables.

2.4 Abandoned ideas

2.4.1 Stereo Camera

Before coming with the idea of using a table-level coordinate system, the team explored with a stereo camera approach to find the distance between persons and tables. A low-cost implementation of a stereo camera is tested by placing 2 identical cameras parallel to each other shown in Fig 2.4.1.1.

![Parallel placement of 2 identical cameras to work as a stereo camera](image)

The steps of stereo matching are shown in Fig 2.4.1.2. The 2 cameras will first be calibrated separately using chessboards. Afterwards, using the camera matrices produced from the individual calibrations, stereo calibration will be performed, followed by stereo rectification and stereo matching. Disparity value of pixels that are successfully matched will be returned.
The pair of cameras captured the scene and produced a disparity map in Fig 2.4.1.3. 2 tables and the ceiling can be recognized from the disparity map. Areas lighter in color are objects closer to the camera. Areas in black are pixels which the cameras failed to calculate the disparity. Mapping the disparity information back to the original color image, an 3D image was constructed shown in Fig 2.4.1.4 which reflects the actual distance between the 2 tables and the ceiling.

The 3D image only contains objects that are within around 8 meters distance, indicating the valid range of the set up is around 8 meters. Using this result, at least 4 sets of stereo cameras are required to cover the entire room. It would be infeasible to implement this on 3/F because objects could span across the field of view of multiple cameras. This approach came with a low feasibility and complicated computation, and thus, not pursued.
Chapter 3 Result and Findings

3.1 4/F

3.1.1 Human Detection with pre-trained detection model

The pre-trained human detection model produced unsteady results which depends on the movement of the visitor. The model is able to detect persons from the top down with a few occasional misses. Please see Fig 3.1.1.1-3.1.1.2.
Shown in Fig 3.1.1.2, the model failed to detect the person at different kinds of poses.

Fig X.X Human detection output in which the person sitting at the bottom left corners without placing the hands on the table was not detected. The instability is understandable that when important features of a person are missing from the camera. With a refresh rate of around 0.5 seconds, the table status tends to flicker.

The inability of detecting every pose of a person hinders the status reliability. Nevertheless, when processing some testing videos, such as videos to be discussed in Section 3.3.3, the model is able to detect the person at least once during his stay at the seat. Therefore, the detection misses of the model was considered tolerable. A delay on status update on seated status could be imposed to lessen the impact.

### 3.1.2 Canny edge detection to look for hogged tables

A threshold was set at 100 and tables that enclose more than 100 edge pixels were labelled as empty. Empty table have number of edge pixels ranging from 0 to 70. Occupied tables have value starting from 130. During the testing of 200 images with no seating status considered, the result achieves 100% accuracy. The accuracy of using edge detection to check Hogged statuses is very high.

Below is a list of observations when placing various objects on the table.

<table>
<thead>
<tr>
<th>Object</th>
<th>Number of edge pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>No objects</td>
<td>0-70</td>
</tr>
<tr>
<td>Pen</td>
<td>132-135</td>
</tr>
<tr>
<td>Printer notes</td>
<td>466-185</td>
</tr>
<tr>
<td>Mobile phone</td>
<td>562-573</td>
</tr>
<tr>
<td>Black jacket</td>
<td>763-1025</td>
</tr>
<tr>
<td>Book</td>
<td>1009-1036</td>
</tr>
<tr>
<td>Laptop</td>
<td>1113-1289</td>
</tr>
</tbody>
</table>
Two examples are illustrated in Fig 3.1.2.1 and 3.1.2.2. Red means Seated, yellow means Hogged and green means Empty. The digits in white shows the number of edge pixels within the table. Seats that are hogged have number of edge pixels larger than the threshold. Seats that are empty have number of edge pixels smaller than the threshold.

![Fig 3.1.2.1. Table status result and number of edge pixels within the table (left) and the original image (right)](image)

![Fig 3.1.2.2. top right desk: The number of edge pixels is 133 when a pen is put on the table](image)

### 3.3.3 Overall Performance

It is a challenging task to evaluate the accuracy of the system that output classification results for a video stream. The team proposed an evaluation method to reflect the actual sensitiveness and losses of the system.

3 testing videos were sampled for evaluation. Each video clip was processed by the system to produce an output video. Whenever the system obtains a new set of data from the database for every half second, the detection output refreshes. Each cycle is processed for approximately 0.6 seconds by the GPU machine. The GPU machine the team used for this project is NVIDIA GeForce 930M with 4GB of memory. The output result was crosschecked for every refresh.
In each sample, only seats that contain scene movements or human movements were selected as valid sampling points. Seats that are static, or have no scene change from the previous video frame were ignored. This is to avoid bias and overly counting the number of correct scene detection when the seat has no movement. In comparison, seats that were seated by a person were selected for the entire period because different poses of the person affect the human detection result, hence the final status output.

Examples of scene movement or human movements that were selected:
- The first frame right after a person leaves an object on the desk
- The first frame right after a person leaves the seat
- The first frame right after an object is removed from the seat
- The starting frame of the video
- Every frame when a person is at the seat
- Every frame when a person is walking behind a seat

Examples of static scenes that were not selected:
- Second frame and onwards of an object being placed on the desk
- Second frame and onwards of a seat being left empty

Below are the videos used for evaluation:

<table>
<thead>
<tr>
<th>Video Name</th>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2_evaluation</td>
<td>Second frame and onwards of an object being placed on the desk</td>
<td><a href="https://youtu.be/Vt38cBrIwSE">https://youtu.be/Vt38cBrIwSE</a></td>
</tr>
<tr>
<td>4f_vid1 (starts at 0:23)</td>
<td>Second frame and onwards of a seat being left empty</td>
<td><a href="https://youtu.be/luw1ts0RYdU">https://youtu.be/luw1ts0RYdU</a></td>
</tr>
<tr>
<td>hog_evaluation (starts at 02:17)</td>
<td></td>
<td><a href="https://youtu.be/nf4tpXnsQLE">https://youtu.be/nf4tpXnsQLE</a></td>
</tr>
</tbody>
</table>
The wrong identifications from Table 3.3.3.2 could be explained. There is a significant amount of Seated seats wrongly identified as Hogged. As explained in section 3.1.1 on the human detection model, the detection results was not very steady and could miss the person whenever a person changes movement. Those Seated seats that were wrongly identified as Empty were also suffering from the same problem.

There is also a certain amount of Hogged seats wrong identified as Seated. An example is in Fig 3.3.3.2. When a person is passing by at the back, the corners of the bounding box fell into the predefined seating space. This implies that the configuration of seating space corners affects the result significantly. The same goes to the case when an Empty seat is wrongly identified as Seated.
The model was able to produce correct classifications in some instances of the following use cases in Table 3.3.3.3.

<table>
<thead>
<tr>
<th>Seated</th>
<th>Hogged</th>
<th>Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td>About to take a seat</td>
<td>Hogged with laptop</td>
<td>Ordinary empty seat</td>
</tr>
<tr>
<td>Hands off table</td>
<td>Hogged with jacket</td>
<td>People passing by</td>
</tr>
<tr>
<td>Lying on the table</td>
<td>Hogged with paper</td>
<td></td>
</tr>
<tr>
<td>Standing up</td>
<td>Hogged with small white object</td>
<td></td>
</tr>
<tr>
<td>Turning sideways</td>
<td>Hogged with small black object</td>
<td></td>
</tr>
<tr>
<td>Seated using laptop</td>
<td>Hogged with a white book</td>
<td></td>
</tr>
<tr>
<td>Playing mobile phone</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3.3.3. Use cases for 4/F detection system
Fig 3.3.3.4 Correct Seated status at Top left table: hands off the table (v2_evaluation 00:06)

Fig 3.3.3.5 Correct Seated status at Top left table: lying on the table (4f_vid1 01:05)

Fig 3.3.3.6 Correct Seated status at Bottom right table: standing up (v2_evaluation 00:34)

Fig 3.3.3.7 Correct Seated status at Bottom right table: turning sideways (v2_evaluation 01:37)
Fig 3.3.3.8 **Correct Seated status** at Top left table: using laptop (v2_evaluation 01:11)

Fig 3.3.3.9 **Correct Seated status** at bottom right table: playing on the phone (hog_evaluation 02:45)

Fig 3.3.3.10 **Correct hogged status** at top right table: Hogged with laptop, top left table: hogged with jacket (4f_vid1 01:16)
Fig 3.3.3.11 Correct hogged status at top right table: hogged with paper (hog_evaluation 03:52)

Fig 3.3.3.12 Correct hogged status at Top right table: hogged with a small white object (pen) (hog_evaluation 03:00)

Fig 3.3.3.13 Correct hogged status at Bottom right table: hogged with a small black object (hog_evaluation 00:59)

Fig 3.3.3.14. Correct hogged status at Top right table: hogged with a white book (hog_evaluation 03:31)
3.2 3/F

3.2.1 Table detection by custom training

The first custom training was conducted on 100 images with a 7:3 split of training and testing set. Figure 3.2.1.1 shows the Overall mAP of the model throughout the first custom training.

The mean average precision (mAP) is the metric used to evaluate the accuracy of the detection model. mAP increases with the number of training iterations until it reached a plateau where the accuracy of the model is no longer improving.

In Figure X.X, mAP rises with the number of training iterations and stays almost flat after reaching 0.4. The training was stopped after 18859 iterations when the mAP value failed to increase or decrease. The mAP value indicates that the model produces some mistakes and inaccuracies.

The low mAP value in the first custom training suggests there is insufficient training data. Therefore, the final training was conducted on 900 images with a 10:1 split of training and testing set. The 900 images were collected from 4 filmings on the 3/F plus pictures taken at the human-eye level.
Figure X.X shows the overall mAP of the model throughout the final custom training.

In Figure 3.2.1.2, the overall mAP rises with the number of training iterations and stays almost flat after reaching 0.8. The training was stopped after 17590 iterations when the mAP value failed to increase or decrease.

The mAP for images of different sizes are also significant in showing whether the model is able to detect tables correctly regardless of their distance from the camera.

Figure 3.2.1.3 shows the mAP of the model on different table sizes.
In Figure 3.2.1.3, the overall mAP rises with the number of training iterations. The plateau of the mAP of large, medium and small object sizes is at approximately 0.8, 0.82 and 0.65 respectively. This indicates that the model is weaker in detecting tables that are far away.

Figure 3.2.1.4 to 3.2.1.5 are some output examples of the model on testing image set during training.
Figure 3.2.1.5. are the output examples of the model on an unseen video captures of 3/F.

Fig 3.2.1.6. Video capture of camera 1 (left) and the detection result (right)

Fig 3.2.1.7. Video capture of camera 2 (left) and the detection result (right)

Fig 3.2.1.8. Video capture of camera 3 (left) and the detection result (right)

Fig 3.2.1.6 – 3.2.1.8 are the result of the table detection model. The model was able to detect unoccupied tables that are near to the camera correctly. However, there are some cases which the performance was not as expected. The results are summarized below with Fig 3.2.1.9 – 3.2.1.13.

1. Tables that are blocked by a person in the middle
2. Tables that are blocked by multiple persons
3. Cropping tables placing close together

Fig 3.2.1.13 Multiple detection boxes over cropped tables placing close together

Fig 3.2.1.9 - Fig 3.2.1.13 shows that the performance fluctuates when there are people blocking part of the view of the table. There is a possibility that when a person is standing in front of a table which blocks the table mid-way, the model only identifies a part of the table. There is also a possibility that when multiple persons block a large part of a table, the model is unable to identify the table.

Fig 3.2.1.13 shows that when cropped tables were placed close to each other, the detection model outputted multiple boxes. The result after non-max suppression is still indicating the two tables as one single table.

It is believed that the model suffered from slight overfitting due to the limited training set. The model identifies empty tables very well. When processing tables with objects or persons, the performance is not as expected.

One limitation is that the training image set for the model is not large enough. Due to privacy concerns, the team could only film and collect training data before the library opens. The variety of clothing, lighting from windows, hogging habits and patterns were very limited. It is also likely that many unexpected scenarios or events were not captured. The most ideal way to train the model would be to use real clippings across hours and days.

**3.2.2 Masking and hull shape comparison to look for hogged tables**
Initially, the team planned to use machine learning to determine whether a table is hogged. Due to constraints in filming and the limited number of hogging objects and styles, only a small set of hogging image set was gathered. The model was unable to produce a result better than random guessing.

After a long investigation, it was found out that the reasons causing the inaccurate result is caused by whether the view of the table is blocked by a person. It is difficult to fit a small set of images that had very different features to a single machine-learning model. Using separate trees for blocked tables and unblocked tables would be the way to work on the problem. However, due to the time limitations in arranging another filming, the project team was stuck with a small image set that have to be further segmented for each tree.

For the hogging decision algorithm on tables not blocked by persons, a compromise was made with a sample set of 160 images, 80 of empty tables and 80 of unblocked hogged tables. The set was fit to a decision tree. The tree is shown in Fig 3.2.2.1. Aware of the possibility of overfitting caused by the small training set, the tree was then fine-tuned manually as illustrated in Fig 3.2.2.2. The tree produces an accuracy of 0.773.

![Decision tree](image)

Fig 3.2.2.1 Decision tree for unblocked hogged tables produced from the small amount of training data. Value = [Hog, Seated]
An example of the classification on tables not blocked by persons is as below.

In Fig 3.2.2.3, the table that was not blocked by any persons was occupied with a bag and other belongings. Putting the total area of white contours over the area of the area enclosed by the yellow outline will find out how much the table surface is visible. A low area to hull area ratio suggests there is higher chance that the table is occupied with belongings. This picture is correctly identified as Hogged.

It was even more challenging to build up an image set for tables blocked by a person due to the limited amount of data. As a result, the decision tree was created from various testings and tunings.
An example of the classification on tables blocked by persons is as below.

In Fig 3.2.2.5 the largest contour, which is the rightmost contour, was chosen to compare with the hull area of the largest contour. Since the table has a smooth surface, the shape of the contour is close to the hull area. There are also three contours due to the person blocking the view. Using the tree in Fig 3.2.2.4, the picture is correctly labelled as Not hogged.

3.2.3 Standing filter with Random Forest Classifier

An image set of 1480 records, 740 from the standing class and 740 from the sitting class, was collected from the Internet. The body joint angles and height-width ratio were extracted from the set and fitted to a random forest classifier. Approximately 300 images from the library cameras were used for evaluation.

The classifier was evaluated by the cross-validated accuracy score and F1 score. F1 score was picked as the metric because it measures both the recall and precision. The reason is that the team believes that false positives and false negatives are equally bad. Whether an empty table is wrongly identified as seated, or whether a seated table is wrongly identified as empty are equally undesirable.

Fig 3.2.3.1 shows the accuracy and F1 score of the classifiers with a different number of trees (n_estimators) and maximum depth for evaluation.
Fig 3.2.3.1 shows the accuracy (solid lines) and F1 score (dashed lines) of the random forest classifier when using different parameters for n_estimators and maximum depth.

In the graph, the accuracy and F1 score of with a different n_estimators decrease when the maximum depth increases which is a sign of overfitting. The accuracy and F1 achieve highest in general among the classifiers with different n_estimators when maximum depth is 5. The classifier with n_estimators = 100 and maximum depth = 5, indicated in purple, has the highest accuracy score of 0.90068 and F1 score 0.88467 among other classifiers. These parameters were chosen.

Fig 3.2.3.2 are the decision made by the classifier on 3/F images.
It is found that even the accuracy score of the classifier reaches 0.9%, the wrong identifications appear at least once during every testing video.

### 3.3.4 Consolidate results from multiple cameras

Fig 3.3.4.2 shows the result from individual cameras in Fig 3.3.4.1. Upon gathering their output, the predefined transformation matrices were used to map them to their respective locations on the floor. Tables that were overlapping with each other were consolidated successfully.

In Fig 3.3.4.1, the table at the right of the camera 1 is the same table as the bottom left table of camera 2. Another table at the bottom right of camera 2 is the same table as the bottom left table of camera 3. Their individual status output were visualized in Fig 3.3.4.2 by filling colors into the estimated table shape produced in the masking step. The table outputs were placed together in Fig 3.3.4.3 by transformation and rotations. Notice that a red table is overlapping with a yellow table, and a yellow table is slightly overlapping with a green table. These overlapping tables were merged and a new status was given shown in Fig 3.3.4.4.
Fig 3.3.4.1. Original images from the camera 1 (left), camera 2 (middle) and camera 3 (right)

Fig 3.3.4.2 Visualization of Status result from system

Fig 3.3.4.3. Consolidation of output across cameras with tables across cameras overlapping with each other
3.3.5 Overall performance

It is a challenging task to access the accuracy of the system on 3/F. Unlike the system on 4/F, the detection result changed in a static scene when the table detection model returns a different bounding box.

The output of the system is depends on components from table detection, to locating the shape of table, to finding distance with persons and filter for hogging.

Below are the videos being evaluated. The window with desk icons on a white background is the result returned from the server.

<table>
<thead>
<tr>
<th>Video Name: 9apr_083001_v2_evaluation</th>
<th>Video Name: 1mar_084500_evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://youtu.be/tEjnCl2qzBg">https://youtu.be/tEjnCl2qzBg</a></td>
<td><a href="https://youtu.be/Ro9nbaKmUsc">https://youtu.be/Ro9nbaKmUsc</a></td>
</tr>
</tbody>
</table>

The refresh time in 9apr_083001_v2_evaluation, unlike 1mar_084500_evaluation, is the actual refresh time because the system does not spend time in rendering the colored tables. The average refresh time is approximately 15 seconds. The GPU machine the team used for this project is NVIDIA GeForce 930M with 4GB of memory.
Different use cases were examined as listed in Table 3.3.5.1. The result were evaluated in Fig 3.3.5.2 – 3.3.5.10.

<table>
<thead>
<tr>
<th>Seated</th>
<th>Hogged</th>
<th>Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple persons sitting around a table</td>
<td>Single item</td>
<td>Empty with a chair blocking the view</td>
</tr>
<tr>
<td>Sitting and facing forward</td>
<td>Multiple items</td>
<td>Empty seat that is spanning across cameras</td>
</tr>
<tr>
<td>Sitting and facing backward</td>
<td>Complicated pattern</td>
<td></td>
</tr>
<tr>
<td>Sitting with lifting arms</td>
<td>A person blocking the view</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3.5.1. Use cases of testing 3/F detection system

Fig 3.3.5.2 Multiple persons sitting around a table (blue) and sitting with different angles (purple) (1mar_084500_evaluation 04:19)
Fig 3.3.5.3 Person with lifting arms (pink) (9apr_0083001_v2_evaluation 02:02)
Fig 3.3.5.4 Person sitting with different angles (pink) (9apr_0083001_v2_evaluation 03:43)

Fig 3.3.5.5 Person sitting and facing backward (pink) (9apr_0083001_v2_evaluation 01:10)
Fig 3.3.5.6 Table is hogged with complicated pattern (1mar_084500_evaluation 04:19)

Fig 3.3.5.7. Hogged seats with a laptop (blue) and multiple items (pink) (9apr_0083001_v2_evaluation 02:02)

Fig 3.3.5.8 Hogged seats being blocked by a person (pink) (9apr_0083001_v2_evaluation 02:27)
Fig 3.3.5.9. Empty seat with a chair blocking the view (9apr_0083001_v2_evaluation 01:53)

Fig 3.3.5.10 Empty seat with a chair blocking the view (9apr_0083001_v2_evaluation 03:23)

Fig 3.3.5.11 – 3.3.5.15 in the following part are examples of incorrect classifications.
In Fig 3.3.5.11, a hogged table outlined in red was wrongly identified as Seated because the sit-stand classifier produced a wrong result into thinking the person is sitting. This is due to the fact that when the person opened her arms, the width to height ratio of the person reduces. This misguided the random tree classifier into thinking the person is in sitting position. As a result, the seat was mislabeled as Seated.
In Fig 3.3.5.12, a seated table outlined in pink was wrongly identified as empty. When looking at the table shape rendered in the result tab, the size of the table is very small. The reason is that a large part of the person was not captured into the image. AlphaPose was unable to locate the person. Therefore, the table was labelled as empty.
Fig 3.3.5.13 Occupied tables were being labelled as Hogged (1mar_084500_evaluation 01:55)

Fig 3.3.5.14. Alphapose was not able to detect two persons in the image.

Fig 3.3.5.13 shows Seated tables outlined in pink and purple were labelled as Hogged. Their Alphapose result in Fig 3.3.5.14 shows that Alphapose was unable to detect the them when a large part of them were blocked by a ladder and another person. As a result, the tables were not labelled as seated.
In Fig 3.3.5.15, the empty table enclosed by the pink rectangle was wrongly labelled as hogged. The is caused by the white color board placed behind the table. The board has the same color as the table and hence affected the estimation of the table shape which the hogging algorithm depends on. As shown in Fig 3.3.5.16, part of the white board was considered a connected part to the table contour. As a result, more black areas appeared within the estimated table shape outlined in yellow. The area of white contours within the hull area was lowered than the actual value. As a result, the decision tree returned the wrong status.

To sum up, despite each component performing well on its own, the overall losses were compounded when one component relies on another. It was disappointing to say that the system was producing a significant of errors, especially at distinguishing empty and hogged seats and deciding whether the person is in sitting position.
Chapter 4: Evaluation / Discussion

4.4.1 Reliability
The performance of each system has been discussed in the respective overall performance.

The vision system on 4/F has high reliability. It only depends on 2 components: edge detection and human detection with the pre-trained network. It was shown that edge detection is an accurate indicator of whether a seta is hogged or empty. The network is the only component that is less stable in producing a correct human detection in every image frame. Nevertheless, the detection network was proved to have no problem in detecting the person at least once during his stay at the seat.

The reliability of the vision system on 3/F is questionable as there exists a number of environmental factors and inherent shortcomings of the system that could disrupt the accuracy of the system. For environmental factors, it was shown that placing white cardboards behind the table led to a bad estimation of table shape and caused the system to believe the empty table is hogged. The lighting of the room also impacts the pixel color of the table. This reliability of the system could be compromised because the hogging determination process also only works with a pre-defined range of white colors to extract estimated table surfaces from detection boxes.

In terms of the inherent shortcomings of the system, the weakness of each component has been discussed in details in the previous section. For the system to work optimally, it requires an optimally trained table detection model, accurate output from AlphaPose, a well-tuned filter in filtering standing persons and a well-tuned hogging algorithm. All these components were compromised by the limited scale of testing images obtained. Data collection and parameters experimenting also impact directly on the reliability and is a process which requires time for experimenting.

4.4.2 Scalability
To enlarge the coverage of the vision system on 4/F, more cameras have to be acquired and installed. Also, manual configuration of the location of seating space and table has to be performed at the beginning. After that, the remaining part would be to configure the IP address of the camera for communication. The manual configuration also takes around 1 minute and requires no expert knowledge. Therefore, it is manageable to enlarge the coverage of the vision system.

The coverage of the vision on 3/F can also be enlarged, let say onto the adjacent compartment. However, the transformation and rotation matrices for translating the scene to the floorplan has to be provided. An additional feature could be designed to allow users to drag and drop for the points of transformation. Another thing to note is that depending on the capability of the GPU machine performing the processing, it is likely that a more powerful GPU instance is required to get a result within a reasonable timeframe to process more camera streams. Evaluation and experimenting would be required. Therefore, the scalability of the vision system on 3/F is limited.
4.4.3 Adaptability

The vision system on 4/F was designed to cover tables from a top down approach. As long as the cameras were kept right above tables of the same size, it can be deployed onto another setting.

The vision system on 3/F is not adaptable to a new environment which the tables are not the same as those trained for the table detection system. In other words, it does not work in another setting. It also does not work if the floor or the walls become white in color.

4.4.4 Cost

The camera used in the 4/F vision system costs RMB$200. If the entire aisle on 4/F is to be covered it requires 200 x 9 = RMB$1800. The system also has to run on a GPU machine to produce fast responses. It is possible to run object detection on CPU but the speed will be heavily lowered.

The vision system on 3/F runs on a GPU machine that requires a CUDA-enabled GPU. The cost of GPU machine must be taken into account.

4.4.5 Speed

For the detection system on the 4/F, each cycle is processed for approximately 0.6 seconds by the GPU machine. Nevertheless, the network delay should also be added into account The GPU machine the team used for this project is NVIDIA GeForce 930M with 4GB of memory. 40% of the GPU memory was assigned for the process. The time bottleneck is in the human detection which each image takes around 0.41 seconds to process.

For the detection system on the 3/F, each cycle is process for approximately 15 seconds by the GPU machine. Nevertheless, the network delay should also be added into account The GPU machine the team used for this project is NVIDIA GeForce 930M with 4GB of memory. 40% of the GPU memory was assigned to the TensorFlow session for detecting tables. The rest of the GPU memory is assigned to Pytorch for loading in the network and perform processing. Table 4.4.5.1 shows the time used by different component in a cycle. The timings were retrieved from a sample test.

| Time for rectifying the image | 0.25 seconds |
| Time for initialization | 0.011 seconds |
| Average total time for table detection of 3 cameras | 2.9 seconds |
| Time for loading PyTorch objects for Alphapose | 2.18 seconds |
| Time for Alphapose to run cuda and evaluation | 0.116 seconds |
| Time for performing pose estimation in 3 cameras | 6.54 seconds |
| Time for generating table features for hogging detection for all tables in 3 cameras | 2.94 seconds |
| Time for checking table intersection across cameras | 0.42 seconds |
| Time for processing data and convert to JSON for posting | 0.4208 seconds |

**Total time = 14.66 seconds**
Table 4.4.5.1 Time required in each components of 3/F detection system
Chapter 5: Future Improvement and Recommendations

One future work is to add an administrator page and time logging. Library administrators can make use of how long a particular seat is hogged to issue warnings, enforce removal of objects or simply for data collection. The library currently has a rule which seats being hogged for over 30 minutes are under the risk of being removed. The data logging feature will give the library higher transparency on seat usage and allow them to manage the seats more efficiently.

Another future work is to strengthen the data support when writing the algorithms for 3/F detections. It is important to obtain actual users’ data and their pattern so that the system produces a fairly accurate result in production.
Chapter 6: Conclusion

I have presented the work on the “Find A Seat App”, a web application that shows real-time seat status of various zones in the library. The app displays a floor plan where tables and seats are marked as ‘empty’, ‘hogged’ or ‘seated’.

This report explained the computer vision approach to detect seat statuses on 4/F and 3/F. The logic of the detection system on 4/F is straightforward, which only requires edge detection and human detection to deduce the status. The experiment result shows that the system on 4/F has flickering Seated status due to the fact that the human detection model fails to identify the person from every frame. Nevertheless, the model was able to do so at least once during the stay of the person.

The steps of the detection system on 3/F is more complex. It requires training a table detection model, using PoseFlow from AlphaPose to find elbow pixels, setting up a table-level coordinate system on the table plane, and hogging deduction based on the area of white contours. The experiment result shows that the system on 3/F suffers from a reliability problem due to compound errors across different components. When in comparison, the system on 4/F is relatively more scalable and adaptable than 3/F.

An important part of future work is to implement time logging to enable library administrators to have more information on seat usage. More field tests with actual users should also be conducted if the system has to be deployed in production.
References


